



# Migrating from AngularJS to Angular

---

*ANGULARJS IS THE NAME FOR ALL V1.X VERSIONS.  
ANGULAR IS THE NAME FOR THE ANGULAR (V2.X, V4.X, V5.X, V6.X, V7.X) OF  
TODAY AND TOMORROW.*

This whitepaper helps you educate about the benefits of migrating AngularJS projects over to the Angular platform.



# Table of Contents

<b>Are you considering migrating to higher version of Angular JS? .....</b>	<b>2</b>
<b>Migration – Consideration points .....</b>	<b>3</b>
<b>Migration – Best Practices .....</b>	<b>3</b>
<b>An Illustrative Case Study.....</b>	<b>4</b>
<b>Angular – Architecture .....</b>	<b>5</b>
<b>Angular – Unit Testing .....</b>	<b>6</b>
<b>Migration – Challenges .....</b>	<b>7</b>
<b>Benefits of upgrade .....</b>	<b>7</b>



### Are you considering migrating to higher version of Angular JS?

If you are considering migrating to Angular, here are some points why someone should migrate to higher version.



**Maintainability of code** – Over a period of time, when the application code size gets bigger, bad patterns like tight coupling, duplicate code, long methods, large classes, and more bugs are detected in the code. The code should be refactored in this case. So, we consider about refactoring the code. But, Instead of Code Refactoring in the same version we consider rewriting the code in Angular7.



**Performance** – When the lines of code is increased, the performance would gradually be degraded with the amount of Data handled in the application. Angular has reduced the bundled file size by 60% compared to AngularJS. Thus code generated is reduced which helps to accelerate the application performance.



**Tools** – External tools are used in AngularJS for Testing and debugging. But in Angular, the application can be tested, debugged and the Command line tools are inbuilt. The CLI takes care of Jasmine and Karma configuration for you. You can also unit test an Angular app with other testing libraries and test runners. Each library and runner has its own distinctive installation procedures, configuration, and syntax. Continuous integration (CI) servers let you set up your project repository so that your tests run on every commit and pull request.



**Language** – JavaScript does not provide Type Safety where you can store any value in the variable which would cause confusion. But the Angular uses Typescript for type Safety, where you can switch on the Type Safety to store values based on the variable type. Typescript also supports type inference using TSL (Typescript Language Service).



**Better Architecture** – AngularJS's two-way data binding concept results in delays and bottlenecks for development projects. But it is not in the case of Angular. The Angular uses an architecture based on a hierarchy of components and services which helps in reuse of the elements.



### Migration – Consideration points

**Code base size** – While dealing with bigger code-base it was very difficult to maintain and go through these awful implications in AngularJS. But the introduction of view engine in Angular, where the produced code of components can be reduced up to 60%. The bundles are reduced to thousands of KBs. Typescript is a superset of JavaScript which helps to build more robust and structured code. The code size would be relatively smaller compared to AngularJS.

**Skillsets** – As mentioned earlier, technically speaking typescript is JavaScript++. So, if someone with expert skill with JavaScript can easily migrate to Angular using the Typescript language. Typescript gives you advantages like Optional static typing, Type Inference, Access to ES6 and ES7 features. The more modern the code is, the easier it is to attract new people to the project.

#### **Budget vs cost of doing nothing**

- ❖ The official support for AngularJS V1.7 (latest) will be stopped on July 1st 2021. The new plugins whatever created, will be made to support only Angular. So, it is always better to migrate and make use of the latest plugins.
- ❖ If the application is expected to be used over a longer period of time, then it's worth considering the migration. But if there are adequate time for the existing code base support and if the application is in sun setting mode, then it is not worth doing it.
- ❖ Inconsistent coding increases the risks of delayed launches or elevated costs. Unlike it, consistent coding has several benefits, such as it makes sites easier to use and enables the use of templates or pre-defined code snippets.

### Migration – Best Practices

Migration can be done based on any of the two approaches based on the size of the project.

**Incremental** – It is also known as Modular approach. We gradually migrate module by module. It is better approach for large and complex applications and Where the team size is large. One of the keys to a successful upgrade is to do it incrementally, by running the two frameworks side by side in the same application, and porting AngularJS components to Angular one by one. This makes it possible to upgrade even large and complex applications without disrupting other business. The upgrade module in Angular has been designed to make incremental upgrading seamless.

**Big bang** – it is performed as a whole. It is better for small and medium size projects. Complexity is increased when the projects is medium sized and the team size is small. So, the effort will be more and the return will be lesser compared to the efforts made. Implementation is faster with big bang adoption than other methods. However, it can be hard on users to have



to learn the new system immediately. The method also involves a lot of risk – problems that arise during implementation can be unpredictable, plentiful and serious, and fallback systems may be inadequate to deal with them.

### **An Illustrative Case Study**

#### **❖ Our Client: US based emergency dispatch service**

Over a period of time, one of our client application grows bigger in size and it slows down and the performance is not optimal when we do include more and more new features to the application.

#### **❖ Problem Statement**

The challenges and the problem statement we faced while working with AngularJS are

- Maintainability of code and code refactoring becomes difficult as there is a lot of dead codes and also it takes lot of time for fixing bugs. There is no enforced code structure in AngularJS.
- Some of the plugins which was in AngularJS are in maintenance mode, there is no update for long time.
- The repository was not updated for long time. Active development was happening only in the angular version of plugin.
- Large amount of data and data handling which becomes even more worse in AngularJS.

We had to keep building upon the existing AngularJS application for a few months and work on bug fixes at the same time.

The application's end users expected new features as well as performance improvements.

Client intended to migrate to the latest version of Angular sooner and our team was quite excited with the client's perspective. Working with the most cutting-edge toolset is always an opportunity to grow as professionals and to dive into all the latest solutions.

#### **❖ Approach: Big Bang**

We decided on refactoring the application with latest version of Angular using the big bang approach. To reconcile further development with fixing bugs, we decided that half of the team would work on new features and the other half would re-writing the old components in Angular.



### ❖ **Going Big Bang**

- ❖ The first thing we paid attention to was the architecture, as the main demand of our client was the opportunity to customize the widget so that a customer sees only his/her personalized themes and settings, whereas all the customer styles are kept on repos.
- ❖ After architecture, we thought about routes (URLs), that is, how URLs change while one is surfing the website.
- ❖ We do have all the plugins and UI ready, so we are able to migrate over a short period of time.
- ❖ Due to type-checking feature, we were able to identify errors at an early stage while writing the code.
- ❖ We make use of unit testing capabilities in Angular CLI to improve the quality in code.
- ❖ Data interaction and performance is better compared to that of AngularJS.
- ❖ Build is more simplified.

### **Angular – Architecture**

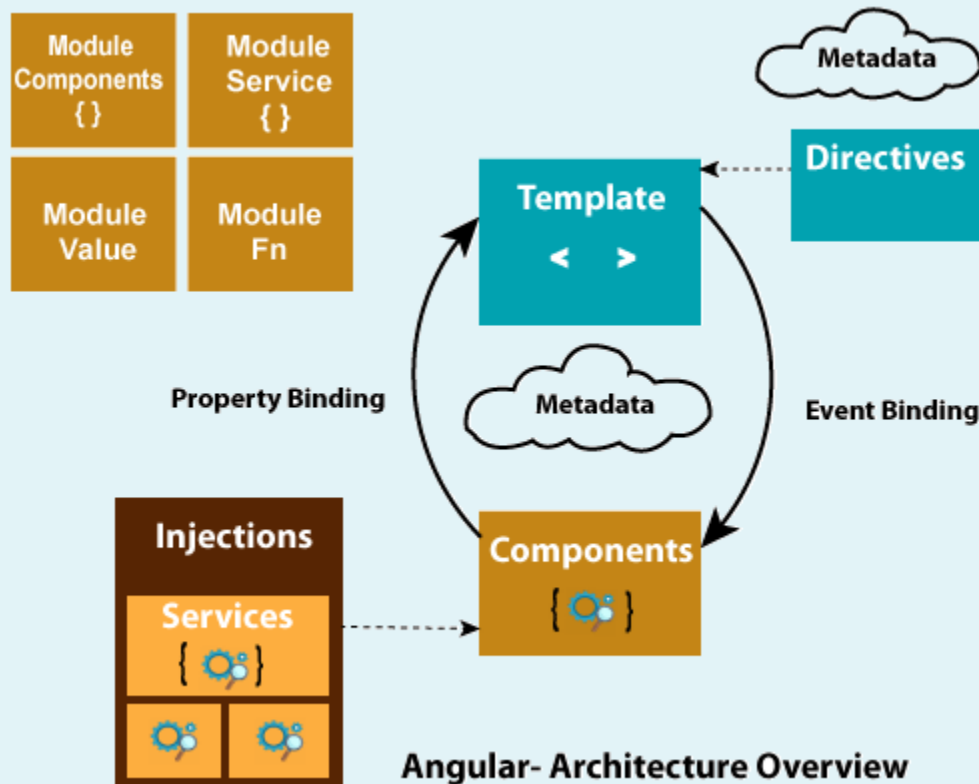
Angular 7 implements core and optional functionality as a set of Typescript libraries which you import in your app.

An Angular 7 app always has at least a root module which enables bootstrapping, and typically has many other feature modules.

- ❖ Components define views, which are the sets of screen elements that are chosen and modified according to your program logic and data by Angular 7.
- ❖ Components use services, which provide specific functionality not directly related to views. Service providers can be injected into components as dependencies, making your code modular, reusable, and efficient.
- ❖ Components and services both are simply classes with decorators that mark their types and provide metadata which guide Angular to do things.
- ❖ The metadata for a component class associates it with a template that defines a view. A template combines ordinary HTML with Angular directives and binding markup that allow Angular to modify the HTML before rendering it for display.
- ❖ The metadata for a service class provides the information Angular needs to make it available to components through dependency injection (DI).



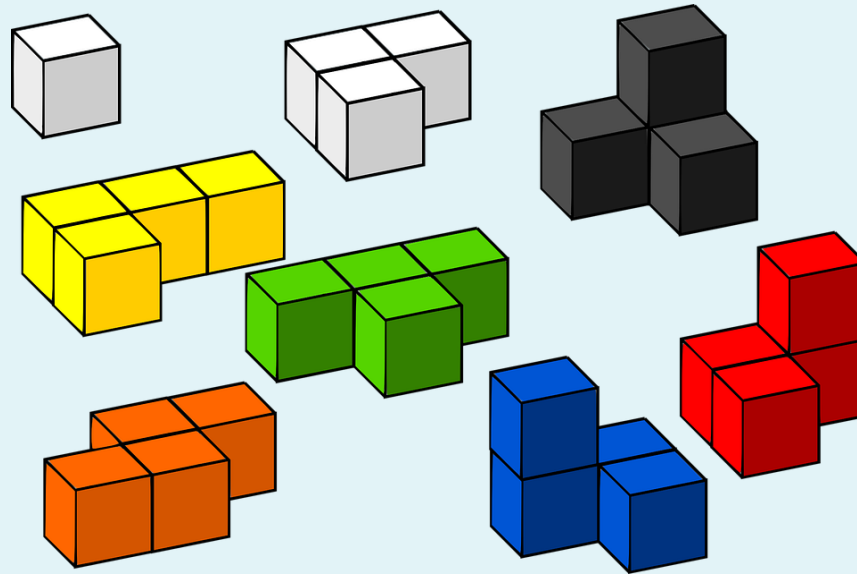
- ❖ Angular 7 NgModules import the functionalities form other NgModules just like other JavaScript modules.
- ❖ NgModules allow their own functionality to be exported and used by other NgModules. For example, if you want to use the router service in your app, you can import the Router NgModule.
- ❖ Event Binding: Event binding is used to bind events to your app and respond to user input in the target environment by updating your application data. Read more about event binding
- ❖ Property Binding: Property binding is used to pass data from component class and facilitates you to interpolate values that are computed from your application data into the HTML.



## Angular – Unit Testing

Have you been looking for a way to test what you have built based on user behavior? I really don't expect you to test each behavior one at a time, as this method does not only waste your time but is also ineffective. Writing tests for different coupling blocks in your application will help demonstrate how these blocks behave.

## AngularJS to Angular migration



One of the easiest ways to test the strengths of these blocks is writing a test for each of them. Writing a test for your blocks (components, services etc) can easily detect when there is a break.

When you create a new project with the cli, a default component and test file are added.

### Migration – Challenges

- ❖ We cannot reuse the same AngularJS code for the migration. The code has to be rewritten when one migrates from AngularJS to Angular.
- ❖ As mentioned earlier, in case if the plugins whichever used for AngularJS is not available for Angular. Then, we have to create new plugins which would support Angular.

### Benefits of Upgrade

- ❖ If we do have the AngularJS code and the UI ready, then it would be easier to understand the logic behind which would help in migration.
- ❖ All the plugins whichever is available for AngularJS, mostly available for Angular also. So, we don't need for search on new plugins.
- ❖ Angular has type-checking feature and reduce runtime errors, which help developers identify errors at an early stage while writing the code. Typescript is going to spit out any compilation error unlike the JavaScript counterpart which is an interpreted language.



## AngularJS to Angular migration



- ❖ Unit test tools comes with the Angular, hence the time taken for configure separate tool is reduced.
- ❖ The official support for AngularJS V1.7 (latest) will be stopped on July 1st 2021. The new plugins whatever created, will be made to support only Angular. So, it is always better to migrate and make use of the latest plugins.
- ❖ When it comes to size of data, Angular usually gives a good performance compared to AngularJS.
- ❖ Page data interaction is better than AngularJS.
- ❖ Angular supports Ionic frameworks, so any one can develop Mobile apps using Angular. It supports Progressive Web Apps where the developer can make use of core features of the mobile (like Notifications, Camera etc).
- ❖ Build is simplified in Angular, which would improve the performance of the application.

## About OptiSol Business Solutions

Optisol is proficient service provider for Social, Mobile, Analytics and Cloud solutions and focusing on building innovation for our clients in more than 20 countries. With 90+ smart employees, we act as the reliable outsourcing services partner, providing optimized business solutions with delivery excellence to both small and medium enterprises worldwide. Through our services on Consulting, Solutions, Mobility, Cloud Computing and Big Data, we assist our clients to transform and thrive in a changing world.

We are equipped with creative engagement models and operations framework that suits outsourcing needs from all gamut of customer's startup, Independent Software Vendor (ISV), small or mid-size companies and Enterprises. It has been serving its customer base across breadth and width of technology outsourcing needs that spans across.